AXP-042-050322

PLUG MAGIC シリーズ PIO アダプタ

AXP-PI02

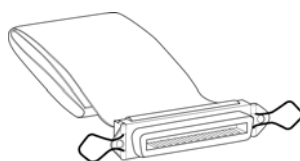
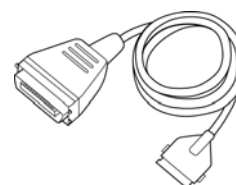
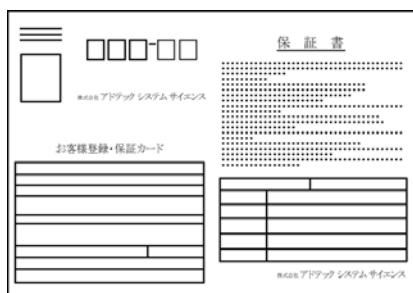
取扱説明書

株式会社 アドテック システム サイエンス

すべて揃っていますか

本体と次の付属品がすべて揃っているか確認してください。
万一、不足の品がありましたらお手数でもお買い上げの販売店もしくは当社までご連絡ください。

同梱品

□AXP-PI02 カード本体

□付属ケーブル

□接続ケーブル

□お客様登録カード･保証書

□サポートディスク

《おことわり》

(1) 本書の内容の一部または全部を無断で記載することは、禁止されております。
(2) 本製品の仕様および本書の内容は、将来予告なく変更することがあります。
(3) 本書の内容につきましては、万全を期して作成いたしましたが、万一ご不審な点やお気付きの点がございましたら、当社までご連絡ください。
(4) 本製品は、出荷の際十分な検査を行い万全を期しておりますが、万一ご使用中にご不審な点がございましたら、当社までご連絡ください。
(5) 本製品につきましては、保証書に明記された条件における保証期間中の修理をもって、当社の唯一の責任とさせていただきます。本製品を運用した結果の影響につきましては、(3)(4)項にかかわらず責任を負いかねます。
(6) 本文中にある会社名、商品名は各社の商標または登録商標です。

目次

はじめに

この度は、PULG MAGIC シリーズ PIO アダプタ AXP-PI02 をお買い求めいただき、誠にありがとうございます。
本製品の性能を十分ご活用いただくため、本書を熟読され、正しい使用法で末永くご愛用いただきますようお願い申し上げます。

ー動作環境ー

■本製品は以下の動作環境でお使いください。

対応パソコン	PC Card Standard TYPEⅡ規格の PC カードスロットを持ったパソコン DOS/V パソコン、NEC PC98 シリーズのいずれの機種も動作可能です。

ご使用前にそのパソコンにPCカードスロットがあるかをお確かめください。

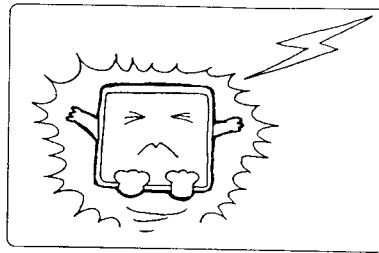
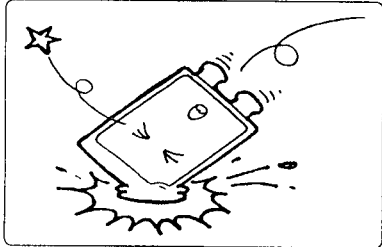
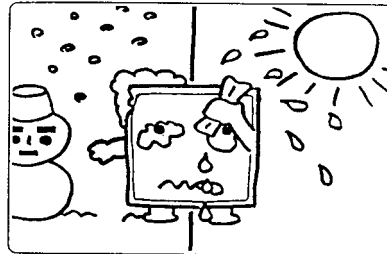
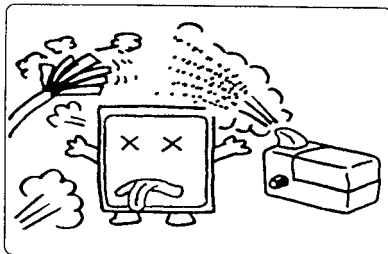
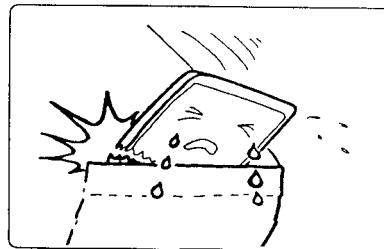
ー特　長ー

- 本製品AXP-PI02は、PC Card Standard TYPEⅡ規格に準拠したPIOアダプタです。PCカードスロット装備の各種パソコンへ装着することにより、簡単にパラレルI/Oの拡張ができます。
- カード内には入出力制御用LSIにPPI（82C55）を1個搭載。24点のTTLレベル入出力を可能としています。
- Windows 95/98/Me/2000/XPに対応。Visual C++、Visual Basic、Delphiに対応したサンプルプログラムにより、効率的にアプリケーションプログラムの開発を行うことができます。

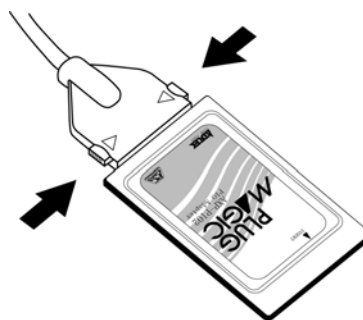
取り扱い上の注意

本製品は非常に精密な電子機器です。お取り扱いに際しては、次の事項を守ってご使用ください。

■このカードはPC Card Standard対応カードスロット以外では使用できません。

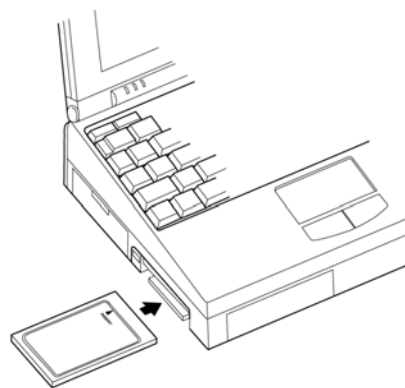
■静電気に弱いので、静電気の起きやすい場所等に放置しないでください。

■本体に衝撃をあたえたり、落としたりしないでください。

■直射日光の当たる場所や低温な場所での使用や保管は避けてください。

■ほこりや湿気の多いところでの使用や管理はさけてください。

■折り曲げ厳禁。破損してしまったカードは修理できません。

1. カードの取り付け

1-1. ケーブルをカードに取り付ける

カードの上面（PLUG MAGICの文字が見える側）と接続ケーブルのカード側コネクタの上面（図のように△のマークのある側）を合わせ、矢印の方向に静かに差し込みます。ロッキング金具の「カチッ」という音がすることを確認します。

接続ケーブルを無理に曲げたり、コネクタとカードとの接続部に無理な力を加えると動作不良や故障の原因になります。

1-2. パソコンへカードを取り付ける

パソコンのカード・スロットにカードを差し込みます。

カードのインターフェース・コネクタ側をパソコンのPCカードスロットに静かに差し込みます。

PCカードTYPE Ｉ スロットには入りません。

PCカードは、誤挿入防止構造になっていますが、無理に差し込もうとすると、パソコンのPCカードスロットやPCカード本体の故障の原因となります。

パソコンの機種によっては、PCカードの裏面を上にし、実装するタイプがあります。ご注意ください。

■カードの取り出し方

PCカードをパソコンから取り出す時は、パソコンのカード・イジェクト・ボタンを押します。カードが少し飛び出します。飛び出した部分を持ち静かに引き抜きます。

ご使用ノートパソコンの取扱説明書カードスロットの項もお読みください。

■電源の ON/OFF 順序

電源を投入するときは、必ずパソコンの電源をONにしてから接続している装置の電源をONにしてください。
また、電源を切る時は、PCカードと接続している装置の電源をOFFにしてからパソコンの電源をOFFにします。

2. ソフトウェアの組み込み

本製品をご使用になる前に、ソフトウェアの組み込み等の準備が必要です。
ソフトウェアは、サポートソフト（添付サポートディスクまたは弊社ホームページ http://www.adtek.co.jp/ からダウンロード）に収められています。
ここでは、サポートソフトを、フロッピーディスク（以下「サポートディスク」）にコピーして使用する場合について示しています。CD-R 等他のメディアをご使用の場合は、適宜読み替えて作業を進めてください。
以下に本カードを MS-DOS で使用する場合、Windows95/98/Me で使用する場合、Windows2000/XP で使用する場合、それぞれについての手順を解説します。

2-1. MS-DOS で使用する場合

ドライバの組み込み方法

MS-DOSのCONFIG.SYSファイルに、以下の行を追加してください。
ただし、カードサービス組み込みの行の後に追加しなければなりません。
カードサービスの組み込みについては、カードサポートソフトのマニュアルをご覧ください。

DEVICE=X:¥PATH¥AXPPI02.EXE[/W1:510][/W2:660]

[ ]のオプションは省略可能です。

ドライバのオプションについて

■デバイスドライバのオプションは、以下の通りです。

I/Oアドレスの指定　「/W1:????」　「/W2:????」
※値は16進数で指定してください。

上記のI/Oウィンドウの指定は、以下に示すI/Oウィンドウの組み（ウィンドウ1とウィンドウ2）を指定するようにしてください。

I/Oアドレス

エントリ	ウィンドウ1（W0）	ウィンドウ2（W1）
0	0100h－0103h	0600h
1	01FCh－01FFh	06FFh
2	0250h－0253h	0700h
3	0300h－0303h	0710h
4	0510h－0513h	0660h
5	05FCh－05FFh	0770h
6	63D0h－63D3h	66D0h
7	65D0h－65D3h	67D0h

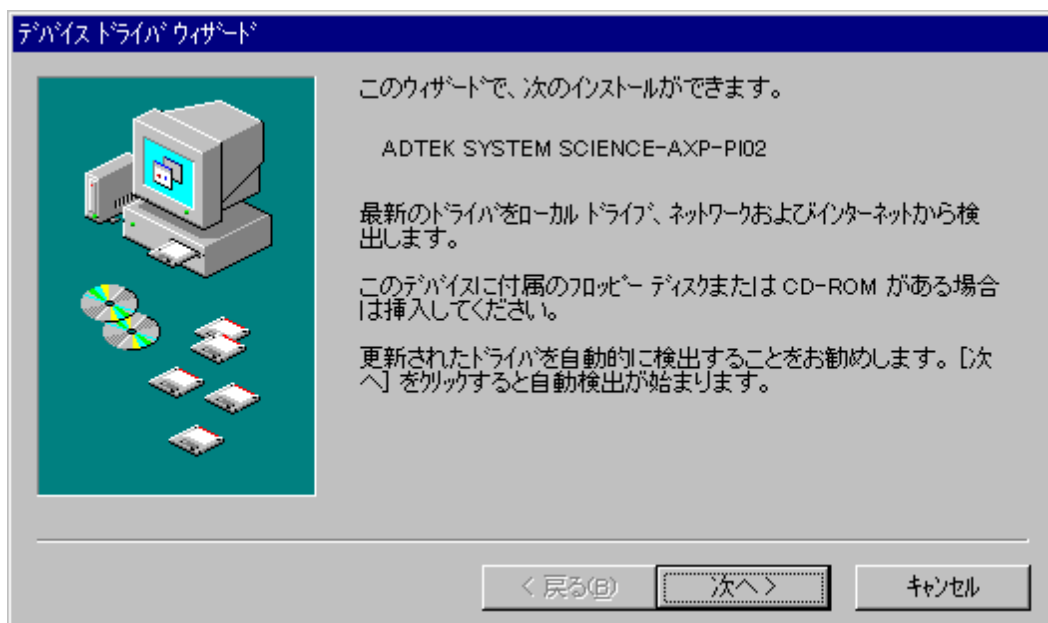
Windows 3.1でご使用になる場合は、MS-DOSと同じ手順でドライバを組み込んでください。

2-2. Windows95/98/Me で使用する場合

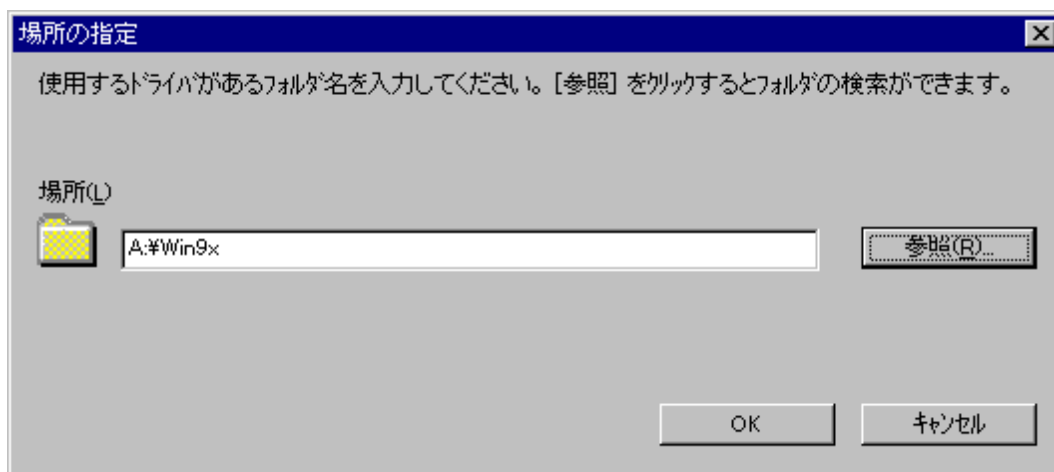
2-2-1. カードの登録

以下は初めて本カードをご使用いただくとき、もしくは登録の削除をした場合の設定です。
この設定は、１度行うと、次回から登録の削除を行わない限り有効です。

ここではAXP-PI02をWindows95で使用する場合を示します。Windows98/Meで使用する場合は、画面の指示に従って適宜読み替えてください。
Windows95が起動したら、PCカードスロットにAXP-PI02を挿入してください。

挿入後、以下の画面が表示されますので、「次へ>」をクリックしてください。

以下の画面が表示されましたら、「場所の指定(O)」をクリックしてください。

サポートディスクを差し込んで、フロッピーディスクドライブを指定します。
ドライブがAドライブのときは以下のように入力し、「OK」をクリックしてください。

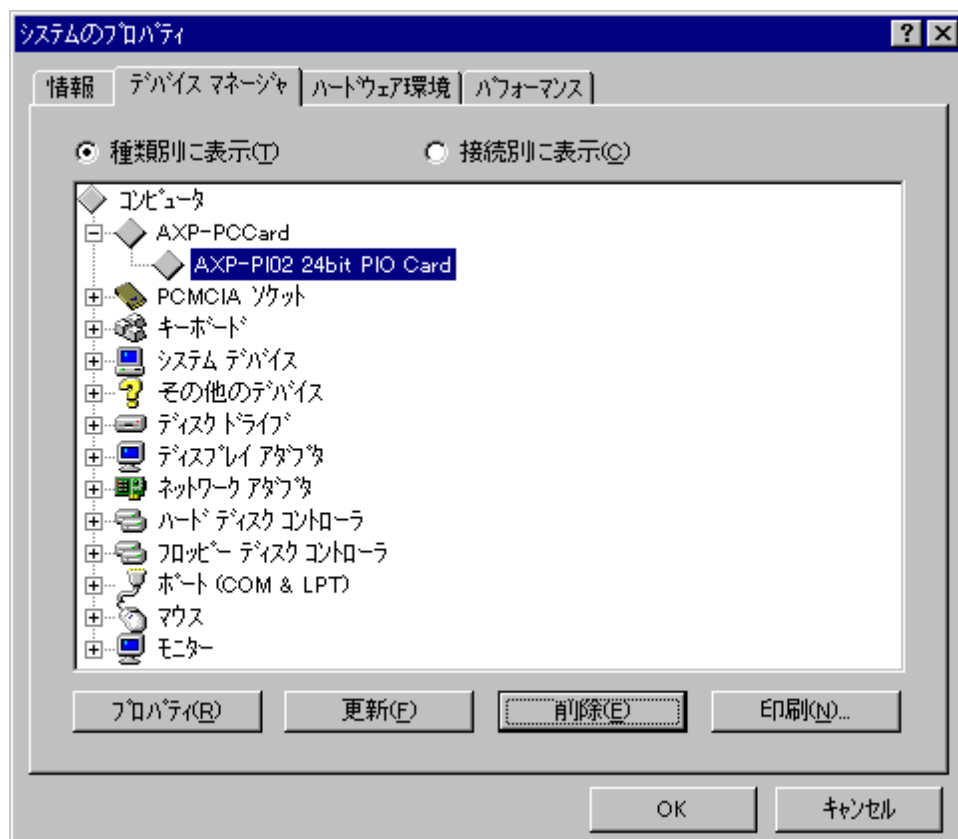
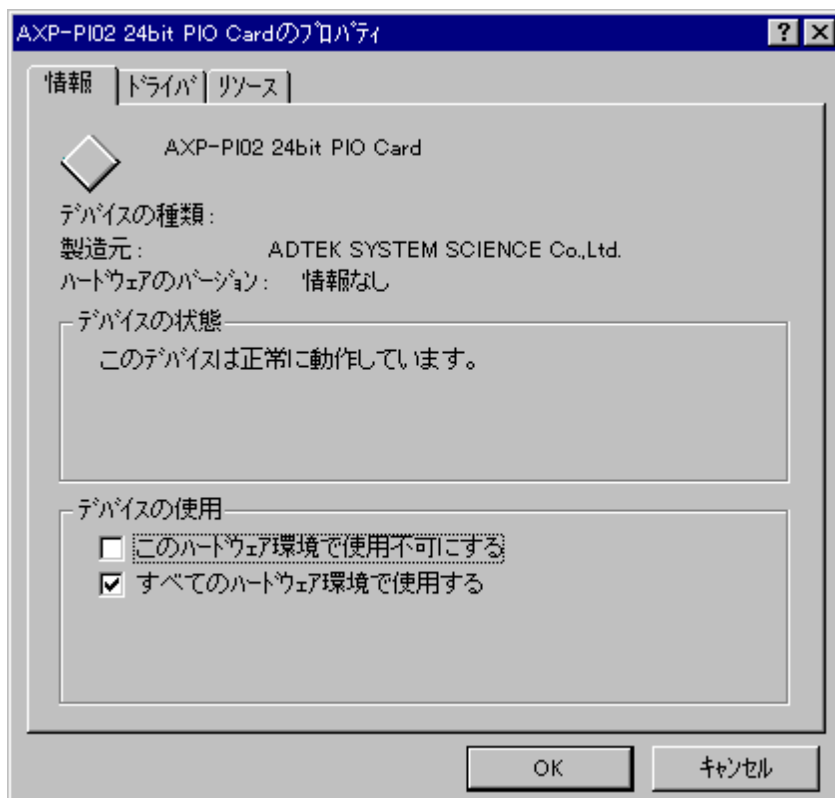
場所の指定
使用するﾄﾞﾗｲﾊﾞがあるﾌｫﾙﾀﾞ名を入力してください。[参照] をｸﾘｯｸするとﾌｫﾙﾀﾞの検索ができます。
場所(L)
A:¥Win9x
参照(R)...
OK
ｷｬﾝｾﾙ

ドライバが見つかったことを確認しましたら、「完了」をクリックしてください。

これでインストール作業は終了です。
数秒後に通常PCカードが挿入されたことを示す認識音（BEEP音）が出ると終了です。
この後、登録の確認を必ず行ってください。

上手くインストールできなかった等、トラブルシューティングはサポートディスクのヘルプおよび、readme.txtをご覧ください。

2-2-2. 登録の確認

カードの登録の確認を行います。
コントロールパネルを開き、「システム」をダブルクリックします。

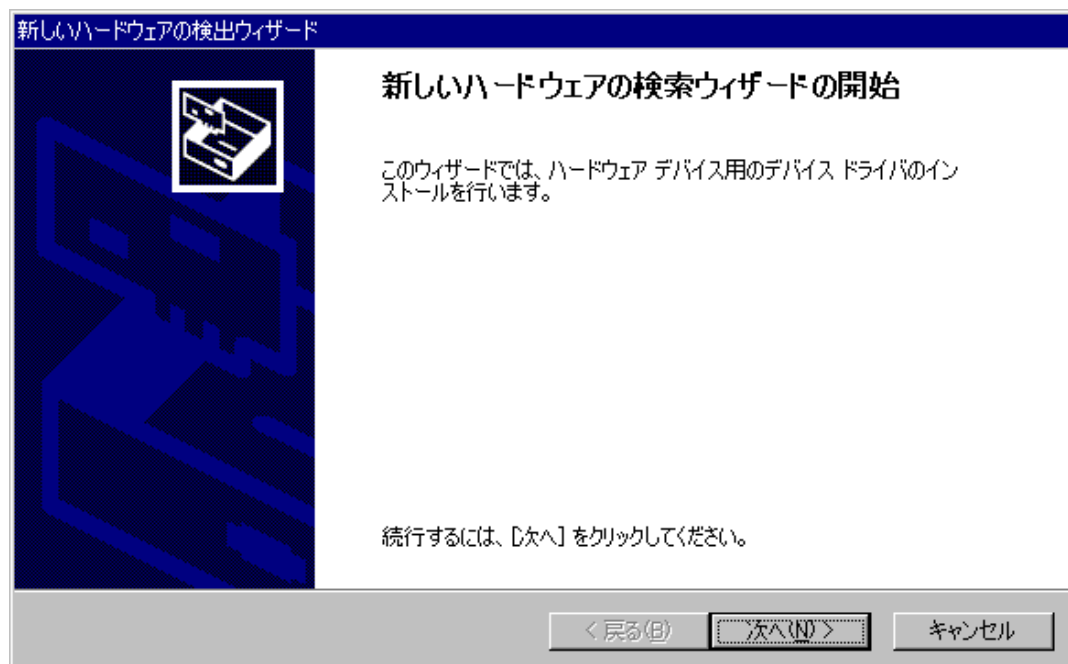
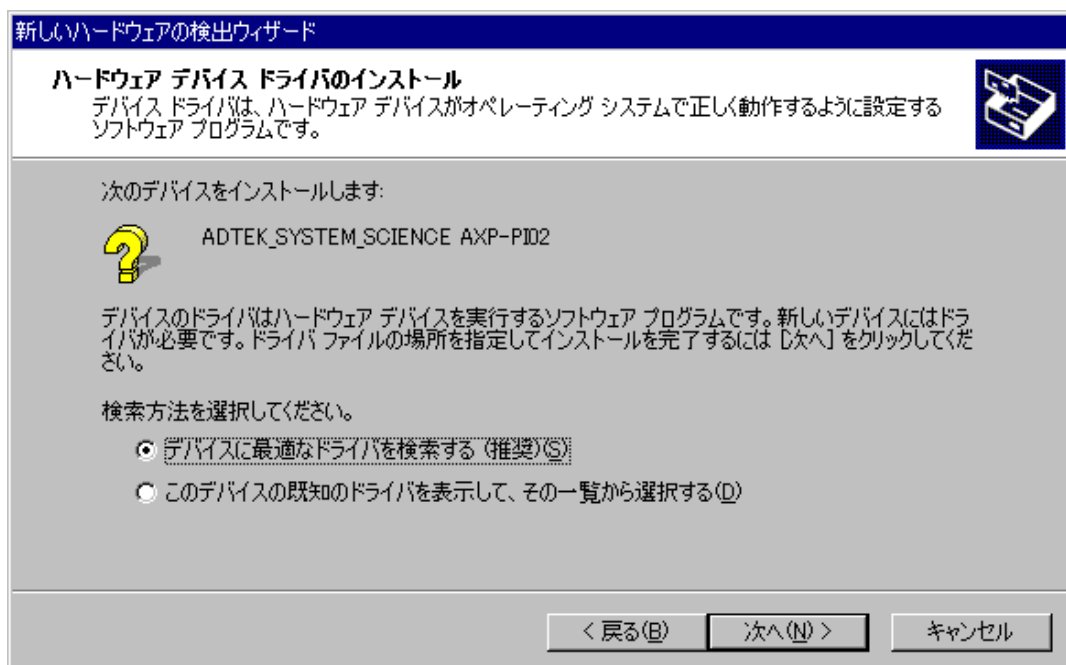
「デバイス マネージャ」のタグをクリックしてください。
「AXP-PCCard」をクリックして、「AXP-PI02 24bit PIO Card」をダブルクリックしてください。

プロパティが表示されましたら、「リソース」のタグをクリックしてください。

画面が表示されましたら、“競合するデバイス”を確認してください。
図のように“競合なし”であれば、正常に設定が完了しています。

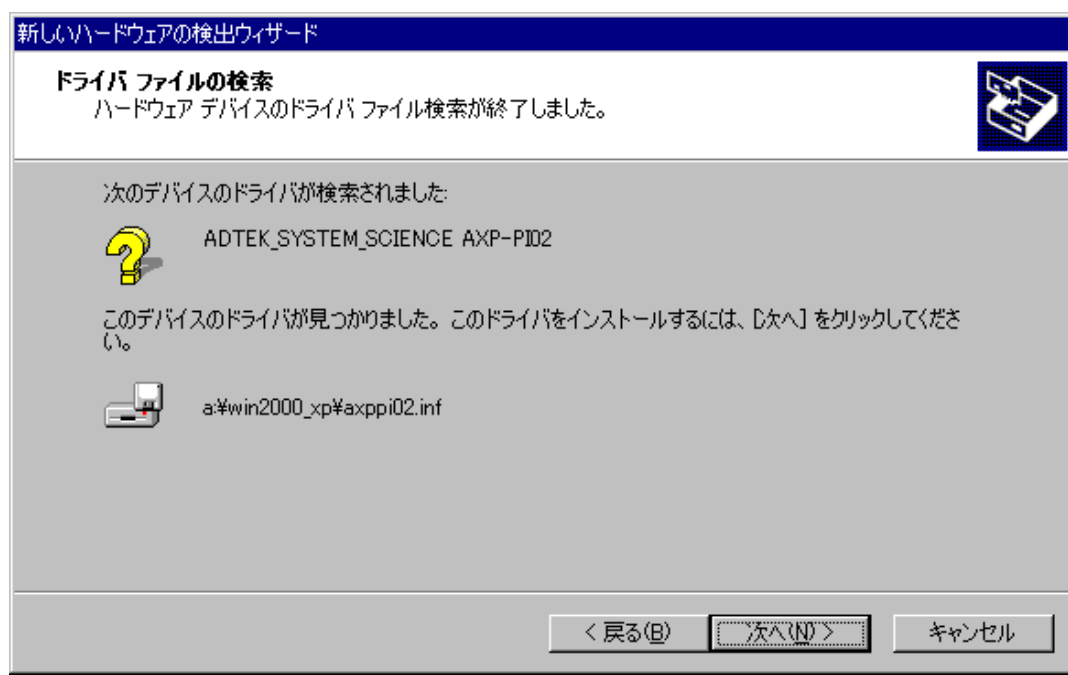
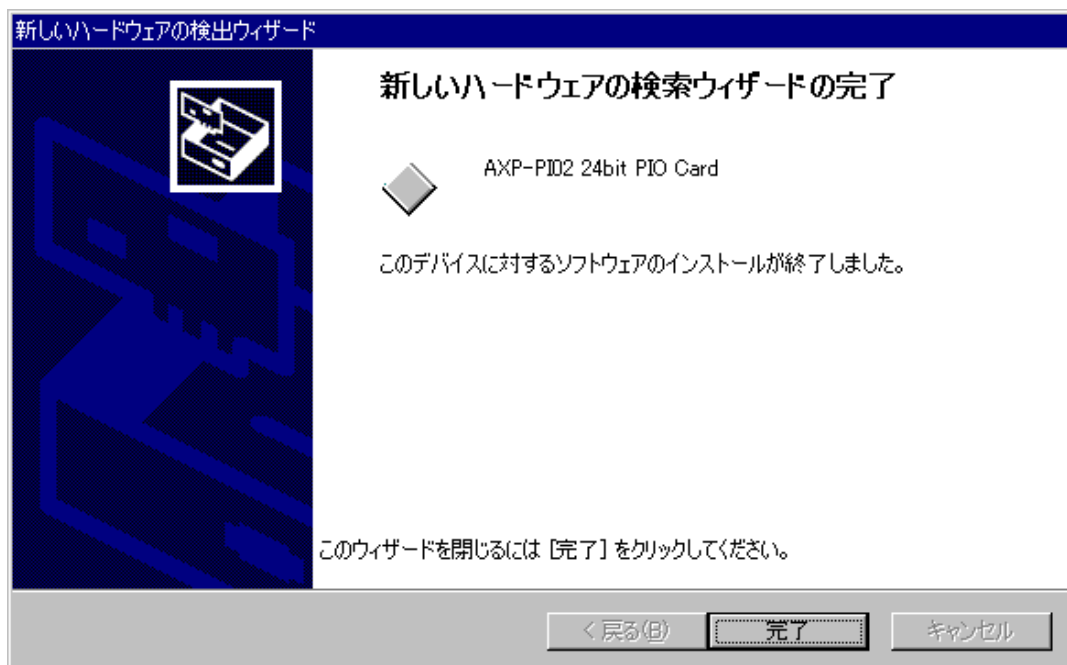
2-3. Windows2000/XP で使用する場合

2-3-1. カードの登録

以下は初めて本カードをご使用いただくとき、もしくは登録の削除をした場合の設定です。
この設定は、１度行うと、次回から登録の削除を行わない限り有効です。

ここではAXP-PI02をWindows2000で使用する場合を示します。WindowsXPで使用する場合は、画面の指示に従って適宜読み替えてください。
Windows2000が起動したら「Administrator」でログオンし、PCカードスロットにAXP-PI02を挿入してください。

挿入後、以下の画面が表示されますので、「次へ(N)＞」をクリックしてください。

「デバイスに最適なドライバを検索する（推奨）(S)」を選択して、「次へ(N)＞」をクリックしてください。

サポートディスクをフロッピーディスクドライブに差し込んでください。
「フロッピー ディスク ドライブ(D)」を選択して、「次へ(N)>」をクリックしてください。

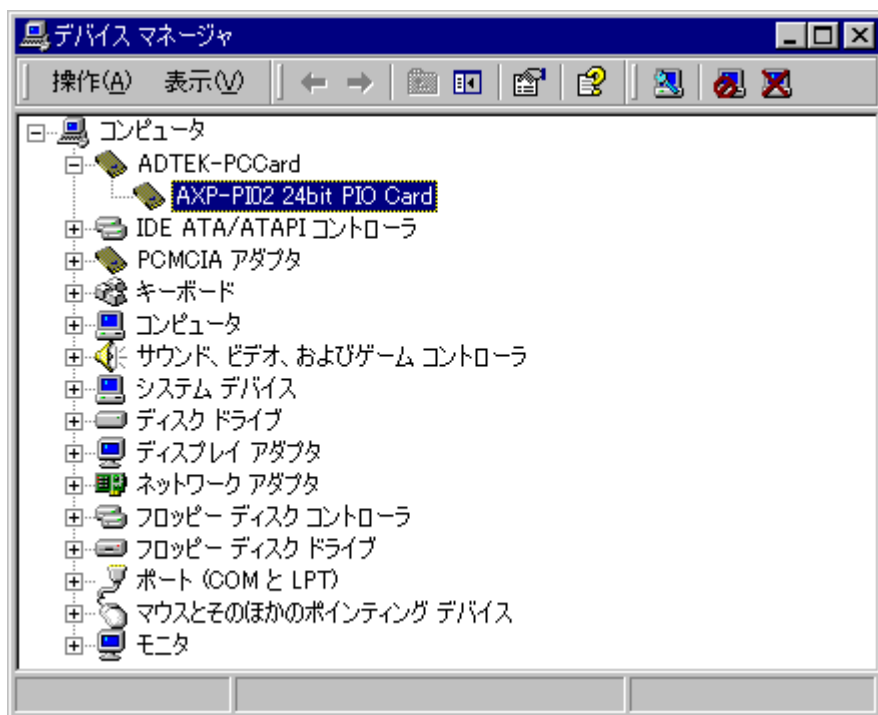
「次へ(N)>」をクリックしてください。

インストールが終了したことを確認しましたら、「完了」をクリックしてください。

これでインストール作業は終了です。
数秒後に通常PCカードが挿入されたことを示す認識音（BEEP音）が出ると終了です。
この後、登録の確認を必ず行ってください。

上手くインストールできなかった等、トラブルシューティングはサポートディスクのヘルプおよび、readme.txtをご覧ください。

2-3-2. 登録の確認

カードの登録の確認を行います。
コントロールパネルを開き、「システム」をダブルクリックします。

「ハードウェア」のタグをクリックして、「デバイス マネージャ(D)」をクリックしてください。

「ADTEK-PCCard」をクリックして、「AXP-PI02 24bit PIO Card」をダブルクリックしてください。

プロパティが表示されましたら、「リソース」のタグをクリックしてください。

画面が表示されましたら、“競合するデバイス”を確認してください。
図のように“競合なし”であれば、正常に設定が完了しています。

3. アプリケーションの作成

アプリケーションプログラムの作成方法を解説します。
AXP-PI02 には、MS-DOS 用のデバイスドライバ、Windows95/98/Me 用のデバイスドライバ（VxD）、Windows2000/XP 用のデバイスドライバ（SYS）、専用ライブラリ（DLL）等がございます。
これらのファイルは、サポートソフトに収めてあります。デバイスドライバ以外のファイルは、作業環境に合わせてコピーしてご使用ください。
また、サポートソフトには、デバイスドライバのアクセス方法や、実際に動作するサンプルプログラムのソースコードも含まれています。
アプリケーションプログラム作成の際に参考にしてください。

3-1. サポートソフトの内容

3-2. 操作手順

AXP-PI02をパラレルI/Oカードとして使用するためには、以下の流れに従って、プログラムを作成します。

1) I/Oウィンドウの先頭アドレス取得

AXP-PI02は、2つのI/Oウィンドウを使用します。

・I/Oウィンドウ1（4バイト）
　ポートA、B、Cとコマンドポート（各1バイト）があります。
　ポートA、B、Cは、AXP-PI02と接続された機器に対しての入出力に使用し、コマンドポートは、カードの動作モードを設定する場合に使用します。

・I/Oウィンドウ2（1バイト）
　このI/Oウィンドウを特にディレクションポートと呼びます。
　ディレクションポートは、AXP-PI02の動作モード設定時のみに使用します。
　これらのI/Oウィンドウの先頭アドレスは、付属のデバイスドライバから取得します。

アプリケーションプログラムを作成する場合には、まずI/Oウィンドウの先頭アドレスをドライバから取得し、それを変数に保存しなければなりません。
以降、入出力を行う時には、このアドレスから目的のポートアドレスを計算します。

ポートA	:	I/Oアドレス1
ポートB	:	I/Oアドレス1＋1
ポートC	:	I/Oアドレス1＋2
コマンドポート	:	I/Oアドレス1＋3
ディレクションポート	:	I/Oアドレス2

2) 動作モードの設定

AXP-PI02のポートA、B、Cに対して入出力を行う前に、各ポートの方向（入力か出力）を設定しなければなりません。
以下のモード一覧表のいずれかのコマンドをディレクションポートとI/Oウィンドウ1のコマンドポートに出力します。

No	I/Oアドレス+0 ポート A	I/Oアドレス+1 ポート B	I/Oアドレス+2 ポート C	ディレクション アドレス	I/Oアドレス +3	ディレクション アドレス	
				①	②	③	④
0	IN	IN	IN	15	9B	15	00
1	IN	IN	OUT	15	92	35	20
2	IN	OUT	IN	15	99	19	08
3	IN	OUT	OUT	15	90	39	28
4	OUT	IN	IN	15	8B	17	02
5	OUT	IN	OUT	15	82	37	22
6	OUT	OUT	IN	15	89	1F	0A
7	OUT	OUT	OUT	15	80	3F	2A
8	IN	IN	－	15	BB	15	50
9	IN	OUT	－	15	B9	19	58
10	OUT	IN	－	15	AB	17	52
11	OUT	OUT	－	15	A9	1F	5A
12	IN	IN	－	15	9F	15	50
13	IN	OUT	－	15	9D	19	58
14	OUT	IN	－	15	8F	17	52
15	OUT	OUT	－	15	8D	1F	5A
16	IN	IN	－	15	BF	15	50
17	IN	OUT	－	15	BD	19	58
18	OUT	IN	－	15	AF	17	52
19	OUT	OUT	－	15	AD	1F	5A

①～④の値を順番に書き込みます。
ポートA、B、Cを表のNo3の状態（入力、出力、出力）に設定するプログラムは、以下のようになります。

out ディレクション ① 15H
out コマンドポート ② 90H
out ディレクション ③ 39H
out ディレクション ④ 28H

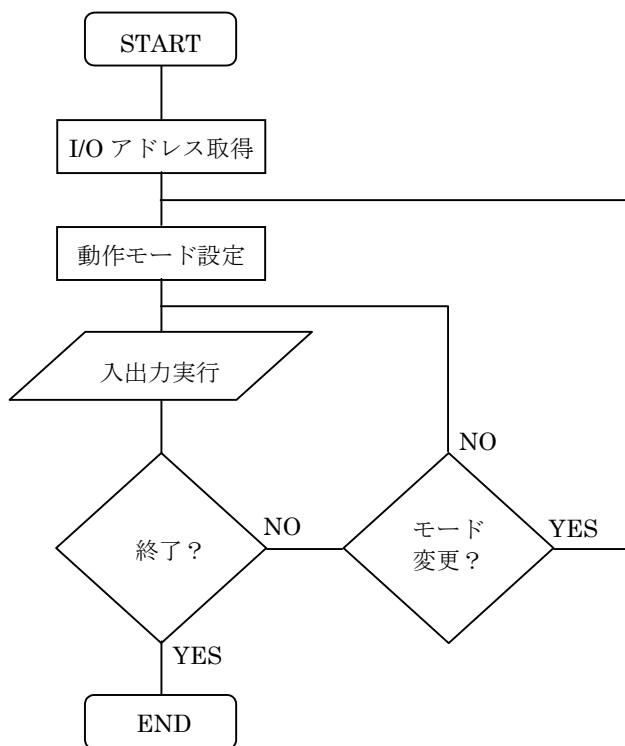
これで、ポートAは入力、B、Cは、出力に設定されます。

表中の①～④の値は、すべて16進数です。

3) 入出力実行

動作モードの設定が終了すると、ポートA、B、Cに対して、入出力が実行できるようになります。ただし、入力に設定したポートに対して出力したり、出力に設定したポートから入力することはできません。
各ポートの入出力方向を変更したい時には、再度「2) 動作モードの設定」を行う必要があります。

4) 入出力の手順

以上の流れをまとめると、本カードで入出力を行う際には下記のような作業が必要となります。

3-3. MS-DOS アプリケーションの作成

MS-DOS上で動作するアプリケーションプログラムの作成方法を解説します。
実際のプログラミングについては、サポートソフト内のMS-DOS用C言語サンプルソースをご参照ください。デバイスドライバへのアクセス方法、動作モードの設定等の処理が記述してあります。
これから解説する処理は、同サンプルのpi02.cファイルに関数としてまとめてあります。
また、デバイスドライバが返す情報のフォーマットも、構造体として、pi02.hファイルに定義してあります。
これらのソースファイルをお手持ちのC言語処理系を使用してコンパイルすることによって、ライブラリとして使用することができます（バージョンアップの際の互換性は保証できませんのでご了承ください）。

3-3-1. I/O アドレスの取得

1） デバイスドライバをオープンする

AXP-PI02用のデバイスドライバをC言語のopen関数を使用してオープンします。
ドライバをオープンする場合は、ファイル名ではなくデバイス名でオープンします。
デバイス名は、"AXP$PI02"です。

```
int hd;
hd = open("AXP$PI02",O_RDONLY)；
if( hd < 0 ){
            デバイスオープンエラー！
}
・・・
・・・ アプリケーションプログラムの処理
・・・
close(hd); /* プログラムの最後で、クローズ */
exit(0);
```

デバイスドライバが正しく組み込まれていない場合、オープンエラーが発生します。
CONFIG.SYSが正しく記述されているかどうか、再度確認してください。

2） I/Oコントロールライトでコマンドを書き込む

デバイスドライバに、I/Oアドレス等の情報を取得するためのコマンドを書き込みます。
コマンドの値は”1”で、長さは2バイト（1WORD）です。
これ以外のコマンドは、受け付けません。
コマンドをデバイスドライバに書き込むには、I/Oコントロールライトを使用します。
I/Oコントロールリード/ライトについては、MS-DOSの解説書をご覧ください。
実際のI/Oコントロールの実行方法については、MS-DOS用C言語サンプルプログラム（pi02.cファイルの_ioctrl関数）をご参照ください。

3) I/Oコントロールリードで情報を読み出す

2)でコマンド”1”を書き込んだ後に、I/Oコントロールリードを実行すると、I/Oアドレス等の情報を読み出すことができます。
デバイスドライバは、以下のフォーマットに従って、情報を返します。

WORD（2bytes）	ソケット番号
WORD（2bytes）	予約
WORD（2bytes）	I/Oウィンドウ数
WORD（2bytes）	I/Oウィンドウ1の先頭アドレス
WORD（2bytes）	I/Oウィンドウ2の先頭アドレス
WORD（2bytes）	I/Oウィンドウ1のサイズ
WORD（2bytes）	I/Oウィンドウ2のサイズ
WORD（2bytes）	メモリウィンドウ数（未使用）
DWORD（4bytes）×4	メモリアドレス（未使用）
DWORD（4bytes）×4	メモリウィンドウサイズ（未使用）
WORD（2bytes）×4	メモリ属性（未使用）
WORD（2bytes）	IRQ番号（未使用）
WORD（2bytes）	IRQアトリビュート（未使用）
WORD（2bytes）	DMAチャネル（未使用）
WORD（2bytes）	DMAアトリビュート（未使用）
WORD（2bytes）	予約

上記の情報のうち、AXP-PI02で有効なのは、以下の項目です。
IRQ、メモリウィンドウ、DMAに関する情報は、無効です。

ソケット番号	・・・	AXP-PI02が挿入されているソケット番号 0から開始
I/Oウィンドウ数	・・・	オープンしたI/Oウィンドウの枚数 AXP-PI02は、2枚のI/Oウィンドウを必要とするので、この値は2固定
I/Oウィンドウ先頭アドレス	・・・	I/Oウィンドウの先頭アドレス 1枚目は、ポートA、B、C、コマンドポート（4バイト） 2枚目は、ディレクションポート（1バイト）
I/Oウィンドウサイズ	・・・	それぞれのウィンドウサイズ 4バイトと1バイト

これらのデバイスドライバの情報は、以下の構造体で定義できます。

```
typedef struct {
        unsigned short      log_socket;                  /* 論理ソケット         */

        unsigned short      card_id;                     /* 予約                 */

        unsigned short      io_num;                      /* IOウィンドウ数       */
        unsigned short      io_addr[2];                  /* I/Oウィンドウ1、2  */
        unsigned short      io_len[2];                   /* サイズ               */
        unsigned short      mem_num;                     /* 予約                 */
        unsigned long       mem_addr[4];                 /* 予約                 */
        unsigned long       mem_len[4];                  /* 予約                 */
        unsigned short      mem_attr[4];                 /* 予約                 */
        unsigned short      irq;                         /* 予約                 */
        unsigned short      irq_attr;                    /* 予約                 */
        unsigned short      dma_ch;                      /* 予約                 */
        unsigned short      dma_attr;                    /* 予約                 */
        unsigned short      status;                      /* 予約                 */
} PI02_INF;
```

この構造体は、MS-DOS用C言語サンプルのpi02.hファイルにあります。

4) I/Oウィンドウの先頭アドレスを取得する

3) で読み出した情報から、I/Oアドレスを取得します。
I/Oウィンドウの先頭アドレス1、2が、AXP-PI02が使用できるI/Oアドレスです。
この2つのアドレスを変数に保存してください。

例)

```
        PI02_INF            inf;
        unsigned short      port_a,port_b,port_c,cmd_addr,dir_addr,dev_cmd;
        union REGS          inreg, outreg ;     /* I/Oコントロール用 */
        struct SREGS        segreg ;

        /* I/Oコントロールで書き込むコマンド */
        dev_cmd = 1;

        /*
        I/Oコントロールライトのパラメータ設定
        デバイスドライバに、コマンド"1"を書き込む
        */
        inreg.h.ah = 0x44 ;         /* ファンクション44H              */
        inreg.h.al = 0x03 ;         /* IOCTRL WRITE                   */
        inreg.x.bx = hd ;           /* デバイスドライバのファイルハンドル*/
        inreg.x.cx = sizeof(dev_cmd);         /* 2バイト              */
```

```
/* コマンドのオフセットとセグメント算出                    */
inreg.x.dx = FP_OFF( (void far *)&dev_cmd );
segreg.ds = FP_SEG( (void far *)&dev_cmd ) ;

/* MS-DOS INT21H実行 */
intdosx( &inreg, &outreg, &segreg ) ;
if( outreg.x.cflag ){
                I/Oコントロール失敗
}

/*
I/Oコントロールリードのパラメータ設定
デバイスドライバから情報を読み出す
*/
inreg.h.ah = 0x44 ; /* ファンクション44H                        */
inreg.h.al = 0x02 ; /* IOCTRL READ                              */
inreg.x.bx = hd ;   /* デバイスドライバのファイルハンドル          */
inreg.x.cx = sizeof(AXP_RES);     /* 読み込む領域サイズ          */
/* バッファのオフセットとセグメント算出*/
inreg.x.dx = FP_OFF( (void far *)&inf ) ;
segreg.ds = FP_SEG( (void far *)&inf ) ;
/* MS-DOS INT21H実行 */
intdosx( &inreg, &outreg, &segreg ) ;
if( outreg.x.cflag ){
                I/Oコントロール失敗
}

/* infからI/Oアドレスを取得する */
port_a    = inf.io_addr[0];           /* ポートA            */
port_b    = inf.io_addr[0] + 1;       /* ポートB            */
port_c    = inf.io_addr[0] + 2;       /* ポートC            */
cmd_addr = inf.io_addr[0] + 3;        /* コマンドポート      */
dir_addr = inf.io_addr[1];            /* ディレクションポート */
```

3-3-2. 動作モードの設定

ポートA、B、Cに対して入出力を行うために、各ポートの入出力方向を設定します。
例えば、ポートA、Bを入力、ポートCを出力に設定する場合のプログラムは、以下のようになります。

```
unsigned int io_addr   → I/Oウィンドウ1の先頭アドレス
unsigned int dir_addr  → ディレクションウィンドウのアドレス

outp(dir_addr, 0x15);      /* ディレクションポートに、15H出力   */
outp(io_addr+3,0x92);      /* コマンドポートに、92H出力         */
outp(dir_addr, 0x35);      /* ディレクションポートに、35H出力   */
outp(dir_addr, 0x20);      /* ディレクションポートに、20H出力   */
```

※MS-DOS用C言語サンプルのset_mode関数（pi02.cファイル）をご参照ください。

動作モード設定後、入出力を行う前に適当なディレーが必要になる場合があります。
ディレーの時間は、使用するコンピュータによって異なります。
処理系によっては、ディレー用の関数がサポートされている場合がありますので、それらをご利用ください。

3-3-3. 入出力実行

動作モードの設定が終了すると、任意のポートに対して入出力を行うことができるようになります。
C言語の場合は、inp/outpなどライブラリ関数で、入出力をサポートしているものでアクセスできます。
入出力関数については、ご使用になるC言語処理系のマニュアルをご覧ください。

Windows 3.1は16bitなので、MS-DOS用のデバイスドライバ（16bit）を使用します。
ソケット・カードサービスの組み込みもCONFIG.SYSで行います。
I/Oアドレスの取得方法､動作モードの設定等は､MS-DOSと同様なので､MS-DOSアプリケーションの作成をご覧ください。

3-4. Windows アプリケーションの作成

Windows上で動作するアプリケーションプログラムの作成方法を解説します。
実際のプログラミングについては、サポートソフト内のWindows用サンプルソースをご参照ください。デバイスドライバへのアクセス方法、動作モードの設定等の処理が記述してあります。
Windows用サンプルソースはVisual C++、Visual Basic、Delphiに対応しています。
axppi02w.*のように「w」が付いているファイルには、DLL内の関数を簡単にコールするためのラッパー（Wrapper）関数が定義されています（DLLのロード/アンロード関数も含まれています）。また、axppi02s.*のように「s」が付いているファイルには、ラッパー関数を使用してデバイスを制御する例が記述してあります。
これらのソースファイルをお手持ちの開発環境に合わせて使用してください。
以下にVisual C++サンプルソースの構築手順の例を示します。その他のサンプルソースの構築手順については、各ディレクトリ内のbuildxx.txtの例をご覧ください。

例) Visual C++サンプルソース構築例

ここではMicrosoft Visual C++ 6.0を使ったサンプルソース構築例を示します。
他の環境をお使いの場合は、適宜読み替えて作業を進めてください。

1. Visual Studioを起動します。別のワークスペースが開かれている場合は、[ファイル]→[ワークスペースを閉じる]で、すべてのワークスペースを終了させてください。

2. [ファイル]→[新規作成]より[プロジェクト]タブの”Win32Application”を選択し、[プロジェクト名]には”Project1”、[位置]に適当なディレクトリを指定し、次画面で”空のプロジェクト”を選択します。

3. 作成されたディレクトリ¥Project1に、全ヘッダ・ソースファイルをコピーし、[プロジェクト]→[プロジェクトへの追加]→[ファイル]でプロジェクトに追加します。

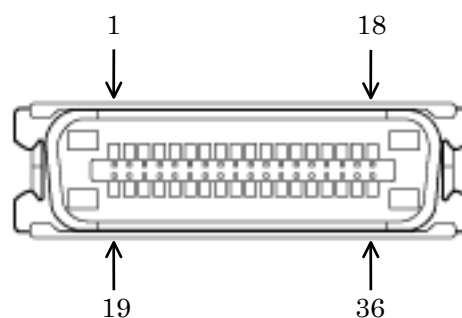
4. [ビルド]→[ビルド Project1]でプログラムがコンパイルされ、実行ファイル”Project1.exe”が生成されます。

実行する前に、あらかじめドライバのインストールとDLLファイルのコピーを行ってください。

4. 外部信号との接続

スイッチや接点または TTL レベルの信号を取り込む時、出力信号で外部機器をコントロールする時は、付属ケーブルを接続ケーブルの入出力コネクタに接続して信号を取り出すか、もしくは市販のコネクタを利用して接続します。
簡単な実験などする時は付属ケーブルの入出力端子に IC クリップなどを取り付けると良いでしょう。

4-1. 信号コネクタ

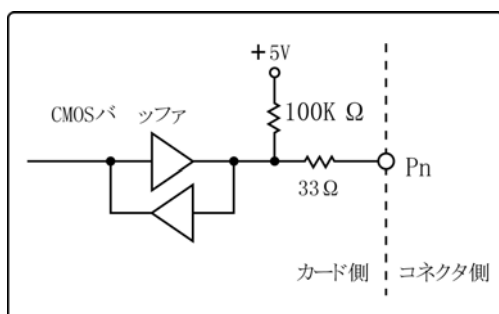
ﾋﾟﾝ番号	信号名	ﾋﾟﾝ番号	信号名
1	PA0	19	PA1
2	PA2	20	PA3
3	PA4	21	PA5
4	PA6	22	PA7
5	PB0	23	PB1
6	PB2	24	PB3
7	PB4	25	PB5
8	PB6	26	PB7
9	PC0	27	PC1
10	PC2	28	PC3
11	PC4	29	PC5
12	PC6	30	PC7
13	GND	31	NC
14	NC	32	NC
15	NC	33	NC
16	NC	34	NC
17	NC	35	NC
18	NC	36	NC

■接続ケーブル側コネクタ
DDK 製 57F-36 又は相当品
■上記に適合する受け側コネクタは
以下の型名のものが使用できます。
○フラットケーブル圧着型
DDK 製 57F-40360-20S
○ソルダカップ型
DDK 製 57-60360

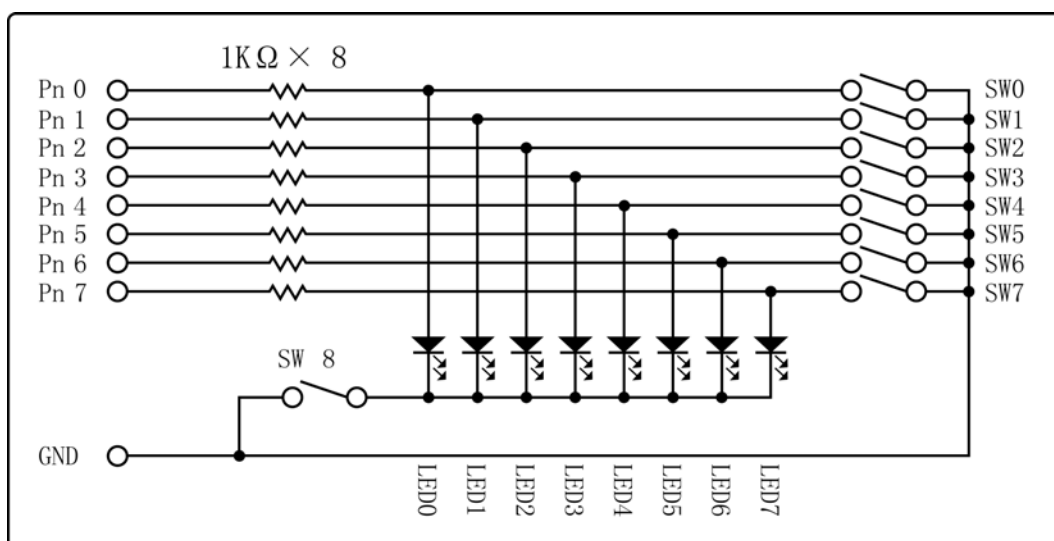
■コネクタ信号表の信号名の意味

・PA0～7： PPI のポート A
・PB0～7： PPI のポート B
・PC0～7： PPI のポート C
・GND： グランド
・NC： 未接続

入出力コネクタのシールドは13番ピン（GND）に接続されています。

4-2. 入出力信号等価回路

入出力信号には特別な保護回路はありません。よって過大電圧を加えたり、出力信号をGND 信号や他の出力信号と短絡（ショートともいう）すると PC カードの故障の原因となります。

※Pn = PA0～PA7, PB0～PB7, PC0～PC7

4-3. 信号接続例

5. 製品仕様

①インターフェース規格		:	PC Card Standard TYPEⅡ準拠
②入出力点数		:	24点（8ビット×3ポート）
③出力時出力電流		:	IOL=IOH=4mA（最大）（100KΩでプルアップ）
④入力時入力電圧		:	HIレベル3.5V（MIN）
		:	LOレベル0.8V（MAX）
⑤使用LSI		:	82C55（モード0、モード1の一部をサポート）
⑥入出力モード		:	付録A参照
⑦割り込み		:	なし
⑧環境条件	温度	:	＋10～＋35℃
	湿度	:	20～80%RH（結露しないこと）
⑨外形寸法		:	幅54(mm)×奥行85.6(mm)×厚さ5(mm)
⑩重量		:	約20g

※82C55は INTEL 8255Aの省電力型で、PPIとも呼ばれます。

※本製品の仕様および外観は製品改良のため予告なく変更する場合があります。

付録 A. 動作モード一覧

サポートしている 82C55 のモードの組み合わせは下表の通りです。

No	I/O アドレス+0 ポート A	I/O アドレス+1 ポート B	I/O アドレス+2 ポート C	I/O アドレス +3	ディレクション アドレス	
				コマンド 1	コマンド 2	コマンド 3
0	ﾓｰﾄﾞ 0 IN	ﾓｰﾄﾞ 0 IN	IN	9B	15	00
1	ﾓｰﾄﾞ 0 IN	ﾓｰﾄﾞ 0 IN	OUT	92	35	20
2	ﾓｰﾄﾞ 0 IN	ﾓｰﾄﾞ 0 OUT	IN	99	19	08
3	ﾓｰﾄﾞ 0 IN	ﾓｰﾄﾞ 0 OUT	OUT	90	39	28
4	ﾓｰﾄﾞ 0 OUT	ﾓｰﾄﾞ 0 IN	IN	8B	17	02
5	ﾓｰﾄﾞ 0 OUT	ﾓｰﾄﾞ 0 IN	OUT	82	37	22
6	ﾓｰﾄﾞ 0 OUT	ﾓｰﾄﾞ 0 OUT	IN	89	1F	0A
7	ﾓｰﾄﾞ 0 OUT	ﾓｰﾄﾞ 0 OUT	OUT	80	3F	2A
8	ﾓｰﾄﾞ 1 IN	ﾓｰﾄﾞ 0 IN	－	BB	15	50
9	ﾓｰﾄﾞ 1 IN	ﾓｰﾄﾞ 0 OUT	－	B9	19	58
10	ﾓｰﾄﾞ 1 OUT	ﾓｰﾄﾞ 0 IN	－	AB	17	52
11	ﾓｰﾄﾞ 1 OUT	ﾓｰﾄﾞ 0 OUT	－	A9	1F	5A
12	ﾓｰﾄﾞ 0 IN	ﾓｰﾄﾞ 1 IN	－	9F	15	50
13	ﾓｰﾄﾞ 0 IN	ﾓｰﾄﾞ 1 OUT	－	9D	19	58
14	ﾓｰﾄﾞ 0 OUT	ﾓｰﾄﾞ 1 IN	－	8F	17	52
15	ﾓｰﾄﾞ 0 OUT	ﾓｰﾄﾞ 1 OUT	－	8D	1F	5A
16	ﾓｰﾄﾞ 1 IN	ﾓｰﾄﾞ 1 IN	－	BF	15	50
17	ﾓｰﾄﾞ 1 IN	ﾓｰﾄﾞ 1 OUT	－	BD	19	58
18	ﾓｰﾄﾞ 1 OUT	ﾓｰﾄﾞ 1 IN	－	AF	17	52
19	ﾓｰﾄﾞ 1 OUT	ﾓｰﾄﾞ 1 OUT	－	AD	1F	5A

表中のコマンド1、2、3はすべて16進数です。

この表は82C55の機能を一部除外しています。除外した設定を制約事項としてまとめると、以下のようになります。

- ポートCは4ビット単位での設定はできません。
- ポートA、ポートBのいずれかもしくは両方をモード1に選択した場合、ポートCのPC4、PC5、PC6、PC7は入力方向に設定していますが、使用できません（コントロール端子のみが使用可能です）。
- 割り込みはサポートしていませんので、割り込みに関する設定/参照は無意味です。また、割り込みの許可/禁止を目的としているビットセット/リセットの機能はできません。
- モード2の設定はできません。

改訂履歴

発行年月日　　1998年11月13日　初版

発行年月日　　2002年08月07日　第2版
Windows2000/XP対応の記述を追加

発行年月日　　2004年07月21日　第3版
付録A、改訂履歴を追加

発行年月日　　2005年03月22日　第4版
お問い合わせに関する情報を修正